W3docs

Photo by Nahel Abdul Hadi on Unsplash

An Ultimate Guide to SSH

W3docs Apr 16, 2020 · 3 min read

SSH keys appear to be commonly used worldwide. The keys are used for executing financial transactions, updating configurations, moving log data, file transfers, interactive logins by system administrators, and for other purposes.

What are SSH Keys?

SSH keys are a credential access that is used in SSH protocol (Secure Shell). Secure Shell is a network protocol that helps you to log in from one computer to another securely, as well as to manage networks,

operating systems, and configurations. Roughly said, SSH keys are nearly the same as passwords, but be sure that it is hundred times more secure to log into a server with SSH keys than use only “hard-to-decrypt” passwords because decrypting SSH is not an easy thing.

SSH keys come in pairs, and each of these pairs consists of a *public key* and a *private key*. These two are types of keys.

1. Public keys (or authorized keys) determine who can access the system.

2. Private (or identity keys) identify users and allow

their access.

When using SSH Key for Git means that you inform Git that your PC is authenticated for that specific GitHub account, and it won't ask you about access ever again because you have already given it your SSH Key.

How to Generate SSH Key?

The steps of creating an SSH Key for Linux and Mac is actually the same as these two operating systems provide modern terminal apps combining with the SSH package.

1. Open the terminal and run:

> *ssh-keygen -t rsa -b 4096 -C “your_email@example.com”*

2. Next, you should choose the file location:

> *> Enter a file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]*

Adding extra security won’t hurt you. So, here you need to add a passphrase to your SSH key.

> *> Enter passphrase (empty for no passphrase): [Type a passphrase]*
>
> *> Enter same passphrase again: [Type passphrase again]*

You can use SSH-agent to save your passphrase securely, so you don't have to re-enter it every time.

Adding New SSH Key to SSH-Agent

The responsibility of the SSH-agent tool is holding the private keys, as well as arranging requests to sign SSH requests with the private keys for security purposes.

1. Run the following code so as to make sure that the SSH-agent code is executing:

> *$ eval "$(ssh-agent -s)"*
>
> *> Agent pid 59566*

2. If it is running, continue the process by adding the new SSH key to the local SSH agent:

```
ssh-add -K /Users/you/.ssh/id_rsa
```

How To Add SSH Key To GitHub Account

1. Your first step should be copying the SSH Key to your clipboard.

2. If the filename differs from the one of the code, match the filename to the current setup:

```
$ sudo apt-get install xclip
```

 # Downloads and installs xclip. If you don't have 'apt-get', you should use another installer (e.g. 'yum')

$ xclip -sel clip < ~/.ssh/id_rsa.pub

 # Copies the content of the id_rsa.pub file to the clipboard

3. The next step is clicking your profile photo in the top right corner of any page and then click Settings.

4. Go to the SSH and GPG keys in the user settings sidebar.

5. Next, click New SSH Key.

6. Type Title and add a description to the SSH Key.

7. Paste your key into the “Key” field.

8. And, finally, hit “Add SSH Key”.

Congratulations, you have successfully added your PC’s SSH key to your GitHub account.

. . .

The crucial part in public-key authentication is that it allows one server to access another server without typing

the password again. This is the reason that is why it gains popularity and is so commonly used for file transfers. The single sign-on operation provides convenience for the users, thus, making it a popular feature.

Check out more git commands and snippets, and get into the Git world to explore new ideas!

Git Github Productivity